SCHEMASESS GRAPH QUERYING

Shengqi Yang, Yinghui Wu, Huan Sun and Xifeng Yan
{sqyang, yinghui, huansun, xyan}@cs.ucsb.edu

OVERVIEW

Motivation
The big graph challenge
Real graph is large.
Real graph is heterogeneous.
- The nodes and relations are from various domains and have rich content.
The query challenge
Queries are often schemaless
- End users possess little or no prior knowledge of the underlying data.
- There is no unified data specification and vocabulary followed by the data contributors and end users.

Contributions
A novel transformation-based matching strategy.
- Name the query and the search engine will do the rest.
An efficient graph search algorithm to fast find the results.
A principled ranking method based on machine learning algorithm.

Impact
◊ I have no idea about schema/data specification/query language; yet I still want to query graph data.
◊ I want to query not only the knowledge graphs but also the document corpus or even the relational tables.

Related Work:
BANKS, YAGO-NAGA, BLINKS, SAGA, NeMa, ...

HIGHLIGHTS

Technique Highlights
- Support various query forms.
  Current: Keyword query, graph query, results visualization and summarization.
  Future: Query-by-example, natural language query, user feedback
- No knowledge on the query language and the underlying data schema is required.

Publications
- Schemasess graph querying - SIGMOD’14 demo, VLDB’14
- Result summarization - VLDB’14
- Ontology-based indexing technique - ICDE’13

The Ranking Model
With a set of matching/transformations, given a query Q and its result R, the ranking model considers

- Node matching: query node v to its match φ(v)
  \[ F_v(v, φ(v)) = \sum a_v f_v(v, φ(v)) \]
- Edge matching: query edge e to its match φ(e)
  \[ F_e(e, φ(e)) = \sum b_e f_e(e, φ(e)) \]

The overall model: a probabilistic model based on Conditional Random Fields (CRFs).

\[ P(R | Q) = \exp\left( \sum_{v \in V} F_v(v, φ(v)) + \sum_{e \in E} F_e(e, φ(e)) \right) \]

Parameter Learning
The parameters \( \{a_v, b_e\} \) have to be determined properly.

- Warm-start: User query logs
- Manual labels
- Cold-start: Automatic training data generation

SEARCHING

Exact search
The transformations incur many match candidates. Exact search is quite expensive.

Inference in the graphical model
- A CRFs model is constructed based on the query and the match candidates.
  - Top-1 result: the most likely assignment (MAE).
  2. Two-level search: sketch graph.

ARCHITECTURE

Framework Architecture
- Online Query Processing
  - Query Prepare: interpret the input query and find the matches from the index.
  - Top-K search: apply the ranking model to find the top results.
  - Logger, Summarizer, etc.
- Back-end modules
  - Indexing: support the transformation based matching.
  - Leaner: train/refine the ranking model with the labeled logs.
  - Distributed scheduler (Akka), etc.

RESULTS

Dataset
- Graph  Nodes Edges Node types Relations Size
  - DBlpedia 3.7M 20M 529 800 40G
  - YAGO2 2.9M 11M 6,395 549 10.5G
  - Freebase 430M 180M 10,110 13,11 384G

Baseline
◊ Spark [Luo07]: IR based ranking/searching method.
◊ SLC: the proposed method in this work.
◊ Unit: a variant of SLC, with equal parameter in the model.
◊ Card: a variant of SLC, with the parameter as the selectivity of the corresponding transformation.

Evaluation

APPLICATIONS

Documents
- Search Portal
- Results navigation
Graphs
- 1. Indexing
- 2. Matching
- 3. User interface
- 4. Visualizing
- 5. Querying
Knowledge bases
- DepdencyGraph
- Graph query answering and MongoDB streaming
- Freenode
- Twitter
- Relational tables
- Graph query answering and MongoDB streaming
- Freenode
- Twitter
- Search Portal
- Results navigation
- 3. User interface
- 4. Visualizing